第十九届“快手杯”北京航空航天大学程序设计竞赛

预赛

2024 年 12 月 5 日 — 9 日

题目解析

A BCPC 2024

显然能作为答案的 y 不超过 2024，直接枚举 y 即可。时间复杂度 $O(2024)$。

B 一句话题面

枚举 z 并计算 $\gcd(x,z)+\gcd(z,y)$ 即可。时间复杂度 $O(y\log y)$。

C I Am the Captain of USTA

注意到 $n\le 30$，$\binom{30}{8}=5852925$，这 8 个人的人选情况数量不会很多。将本题分成两步考虑：先枚举派出的 8 个人的人选，再决定将每个人分到哪条线。容易发现如果存在合法分组方案，答案由 8 人 UTR 之和确定，与第二步无关。

确定人选后，将 8 人按 UTR 从大到小排序，从最大值开始依次匹配相邻的两人，分成 4 组，判断每组是否都合法即可。容易证明这种分组方案是最优的。因此只需 DFS 全部人选，判断合法性，统计 8 人 UTR 之和的最大值。时间复杂度上界为 $O(8\binom{n}{8})$，其中 $8\times\binom{30}{8}\approx 4.7\times 10^7$。

具体实现时可以提前将所有人按照 UTR 从大到小排序，在搜索时进行剪枝优化。当然，不进行任何优化也可以通过本题. 但如果还搜索了每人分别属于哪一组，复杂度过高，会运行超时。

本题限制了人数固定为 8，除了 DFS 之外，还可以通过多重循环枚举的做法通过本题，不过代码实现可能较繁琐。

D 这个题简单，先做这个！

从小到大排列所有偶数，再从小到大排列所有奇数，即 $2,4,6,\ldots,2\times\lfloor n/2\rfloor,1,3,\ldots$。

容易证明此排列可以让 $\gcd(a_i+1,a_{i+1})$ 均取到 1。

E 游戏高手

首先可以发现，长度为 1 的子串价值为 1，所以最终答案一定不会小于 1。由于串 S 由小写英文字母构成，最多只会包含 26 种不同的字符，当子串长度大于 $26^2=676$ 时价值一定小于 1，所以只需要遍历长度小于等于 676 的子串即可。时间复杂度为 $O(26^2|S|)$。

另一种方法是枚举答案中包含的字符种类数 k，使用双指针寻找恰好包含 k 种字符的最短子串。时间复杂度为 $O(26|S|)$。

F 模四识别

模四识别？模式识别！

F.1 方法一

考虑 $dp_{i,j}$ 表示前 i 个数中函数值模 4 与 j 同余的子集数量。转移分为三种情况：当前数作为新子集出现，不选当前数，当前数放入前面的子集中。前两种情况很容易转移，第三种情况假设前面的子集函数值是 x 而当前数是 y，那么新子集的函数值为 $x+y+xy$，直接转移即可。时间复杂度 $O(n)$。

F.2 方法二

注意到

$$\sum_{T\subseteq\{1,\ldots,|a|\}}\prod_{i\in T}a_i=\prod_{1\le i\le|a|}(1+a_i)$$

因此 $f(a)=\prod_{1\le i\le|a|}(1+a_i)-1$，$f(a)\bmod 4=0$ 当且仅当 $\prod_{1\le i\le|a|}(1+a_i)\bmod 4=1$。也就是说模 4 与 0 同余的 a_i 任选，与 2 同余的 a_i 需要选偶数个，其余 a_i 不能选。按模 4 的余数对 a_i 分类，根据数量直接计算答案即可。时间复杂度 $O(n)$。

G 两句话题面

设 $d(x)$ 表示 x 的约数个数，那么 $\gcd(x,z)$ 至多只有 $d(x)$ 种。因此只需分别枚举 x,y 的约数 d_1,d_2，判断是否可行，也即是否存在 $x<z<y$ 满足 $lcm(d_1,d_2)|z$。

由于 10^9 以内整数的约数个数至多有 1344 个，此方法的时间复杂度为 $O(1344^2\log x)$，可以通过本题。

H Good Digits

H.1 方法一

这也是这道题原来的定位：一道难度和代码量适中的搜索题。

当 $n\ge 10^8$ 时，n 位数过多，一定不为好数。但依次检验 10^8 以内的整数仍然会超时。

注意到 $\mathrm{lcm}(2,3,4,5,6,7,8,9)=2520$，可能的余数情况较少。只要确定了 $n\bmod 2520$ 的值，就能确定 $n\bmod 2,n\bmod 3,\ldots,n\bmod 9$ 的值。因此可以枚举 $r=n\bmod 2520$，得到 $r_{n,2},\ldots,r_{n,9}$。再从中任选 1 至 8 个数，枚举这些数位的排列顺序，统计拼成的数中有多少个数 $\bmod 2520$ 的余数等于 r。如果实现得当的话，时间复杂度的上界为 $O\left(2520\times\sum_{i=1}^{8}\binom{8}{i}i!\right)$，其中 $2520\times\sum_{i=1}^{8}\binom{8}{i}i!\approx 2\times 10^8$。

具体实现可以采用深度优先搜索，使用链表加速搜索时的枚举过程；也可以在搜索完最低的 1 至 3 位后根据 $r_{n,2}, r_{n,4}, r_{n,5}, r_{n,8}$ 进行优化剪枝，提高运行效率。当然，不进行任何优化，使用朴素的深度优先搜索或者 `next_permutation` 函数同样能够通过本题。

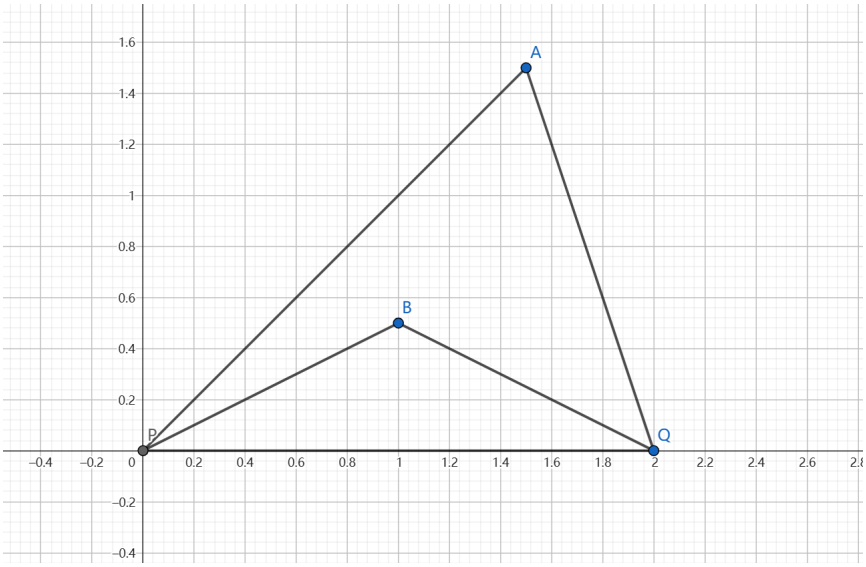
H.2 方法二

这是大家喜闻乐见的做法，*也就是打表*。

具体有两种实现思路：

- 发现好数的总数只有不到 10000 个，提前枚举所有好数，使用二分查找或直接遍历得出有多少个好数位于 $[l, r]$ 内。
- 分块打表，设置一个块大小 B，提前计算出 $B, 2B, \ldots, \lfloor \frac{n}{B} \rfloor B$ 内有多少个好数。只需查表后枚举剩余的不超过 B 个数即可。

I 晶体化



假定我们已经选择了一个魔晶根 PQ，考虑两个凝结核 A 和 B：凝结核 B 在魔晶三角形 PQA 内部或边界上，当且仅当

$$\angle BPQ \le \angle APQ \quad \text{且} \quad \angle BQP \le \angle AQP$$

那么据此，我们重述一个凝结核相对一个魔晶根合法的条件。我们枚举选择的魔晶根 PQ 以后，算出每一个凝结核与 PQ 和 QP 形成的角度 α_i 和 β_i。可以选择的凝结核 i 需要满足：不存在 j 使得 $\alpha_j \le \alpha_i$ 且 $\beta_j \le \beta_i$。要找到这样的 i，只需要把凝结核按照 α_i 排序并扫描，扫描过程中维护 β_i 的最小值。

具体实现上，需要小心地处理 ABP 或 ABQ 共线的情况。此外，由于值域的平方 $(10^9)^2 = 10^{18}$ 已经超过 `double` 能处理的精度范围，最好使用整数坐标和外积实现各种角度判断，以防 `atan2` 产生精度误差影响答案。

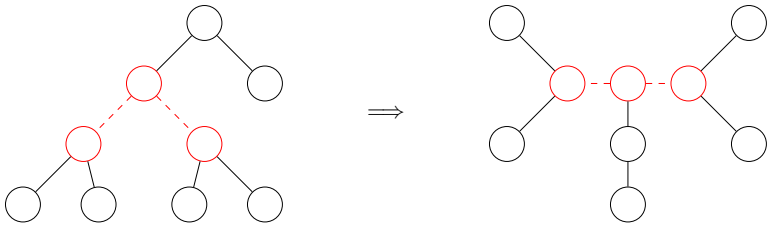
枚举魔晶根的复杂度 $O(n)$，对所有凝结核排序的复杂度为 $O(m \log m)$，总时间复杂度 $O(nm \log m)$。

J 叶的距离

最开始的题目是求一次方的和，后来发现和 CF685B 基本一样，于是改成求平方和，作为一道树上信息统计的练习题放在预赛供大家练手。

结论和原题一致：设 x_u 为子树 u 的最优点，v_i 为 u 的所有儿子，则 x_u 必定是某个 x_{vi} 的祖先或它本身。下面证明这个结论。

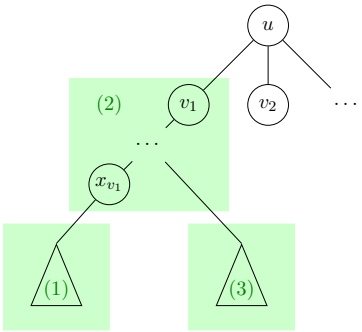
引理 对于某棵子树 u，当 x_u 在子树内任意一条简单路径上移动时，$f(x_u)=\sum_{y\in L(u)} d(x_u,y)^2$ 为单谷函数。



如图，若 x_u 在左图的红色路径上移动，我们可以将这条路径“拉直”为右图，即一条直的链，每个点上挂了若干叶子。

令 c_1 为 x_u 左边叶节点的数量，c_2 为右边叶节点的数量，s_1 表示 x_u 到左边所有叶节点的距离之和，s_2 表示到右边所有叶子的距离和。则当 x_u 从左往右移动时，f 的变化量 Δ 与 c_1,s_1 正相关，与 c_2,s_2 负相关（具体公式可以通过作平方差自行推导），显然 c_1,s_1 单增，c_2,s_2 单减，因此 Δ 单调递增，得到 $f(x_u)$ 在任意一条简单路径上都是单谷函数。

接下来利用引理证明 x_u 必定是某个 x_{v_i} 的祖先或它本身。



如图，(1) 部分为 x_{v_1} 的子树，(2) 部分为 x_{v_1} 到 u 之间的节点（包括 v_1,x_{v_1}），(3) 部分为其它节点。即证：若 x_u 在子树 v_1 中，则 x_u 必定在 (2) 中。

将 $S=\sum_{y\in L(u)} d(x_u,y)^2$ 拆为 S_1,S_2 两部分，S_1 表示子树 v_1 内的叶子节点的距离平方和，S_2 为剩余叶子的和。

若 x_u 在 (1) 中，则必定不优于 x_{v_1}。若移动到 x_{v_1}，S_1 显然减少（x_{v1} 为子树 v_1 的最优点），由于向上移动，S_2 也减少。

若 x_u 在 (3) 中，一直向上跳父亲节点，直到跳到 (2) 中，答案一定变得更优。因为向上跳，S_2 减少。利用引理，构造任意一条经过 (1)(2)(3) 部分的路径，由于 x_{v1} 处取到唯一的谷值，因此两侧都是单调的，从 (3) 到 (2) 的 S_1 的值也一定单调减少（非严格）。

于是证明了 x_u 必定在 (2) 中。

做法：先一遍 DFS 处理出若干需要用到的树上信息。第二次 DFS 时，对于某个 u，先递归计算每个儿子 v_i 的最优点 x_{v_i}，然后从每个 x_{v_i} 尝试向上跳去得到 x_u，直到变化量 $\Delta \geq 0$ 停下。每个点只会被跳到 $O(1)$ 次，每次计算增量也是 $O(1)$ 的，总复杂度 $O(n)$。

难点在于计算增量用到的信息较多，标程对于每个 u 预处理了子树 u 内叶子个数、u 到子树内叶子的距离一次和、平方和、u 到子树外所有叶子节点的距离和。然后通过一系列公式计算 Δ。

K 似花还似非花

考虑点分治，令所有异或和为零的路径在其经过的最浅分治中心处被统计。对于某个分治中心 c，需要考虑的路径 $p(u,v)$ 满足 u,v 处于 c 的不同子树，并且：

$$\bigoplus_{x\in p(u,v)} a_x = \bigoplus_{x\in p(u,c)} a_x \oplus \bigoplus_{y\in p(v,c)} a_y \oplus a_c = 0$$

也就是说对于点 u，能与其配对的点 v 满足 $p(v,c)$ 的异或和是固定值。从 c 出发 DFS，计算 c 到各个点的路径异或和，以及这个点来自 c 的哪个子树。对于某个路径异或和，维护来自不同子树的最小值与次小值，即可快速对 u 求出经过 c 的满足条件的最小的 v。

时间复杂度 $O(n \log n)$。

L 基于 λ 演算的关于 p-进范数下动力系统稳定性的探究与应用

考虑枚举 $\times 2$ 操作的次数 t。对于 $+1$ 操作，如果之前有 i 次 $\times 2$ 操作，那就等价于 $+2^i$。所以不同操作序列的数量相当于将 $y - 2^t x$ 拆分为若干个 $2^0, \ldots, 2^t$ 之和的不同拆分方案的数量。

考虑如何计数。钦定拆分方案按照次幂降序排列，设 $f(i,j,k)$ 表示 2^i 拆分为若干 2 的次幂，最高次幂不超过 j，且最低次幂恰好为 k 的方案数。显然有：

$$f(i+1,j+1,k+1) = f(i,j,k) = \sum_{x=k}^{j} f(i-1,j,x) f(i-1,x,k)$$

将 $y-2^t x$ 写作二进制，从高位到低位考虑是 1 的二进制位，利用 f 的信息转移。具体来说，设 $g(i,j)$ 表示当前从高位到低位考虑到 2^i，当前拆分方案最低次幂为 j 的方案数。初始时 $g(+\infty,t) = 1$。若 $y-2^t x$ 二进制下 2^i 这一位为 0，则 $g(i,*) = g(i+1,*)$。否则：

$$g(i,j) = \sum_{k\geq j} g(i+1,k) f(i,k,j)$$

最终 $\sum g(0,*)$ 即为所需。对于单个 t 可以在 $O(\log^3 n)$ 的时间内完成 DP，总时间复杂度 $O(\log^4 n)$。

Fun Fact：本题题目 ID 为 `classic-problem`。

M Oops

首先我们考虑几个贪心的性质：

1. 最优方案中用到的 ps 一定是从右往左贪心出现的（即从右往左扫，每遇到一个 p 且右边有剩余的 s 就视为一个可用的）。这样是因为选择越靠右的 ps 就能有越多的 o 可用。形式化证明留给读者自己思考。
2. 最优方案中，靠左的 ps 所拥有的 o 数量小于等于靠右边的 ps 所拥有的 o 数量。这点比较好理解，因为靠左的 ps 可用的 o 都是右边 ps 可用的。如果不满足上述条件我们简单交换两者 o 的数量即可。

有了这两条性质之后我们就可以通过一个简单的背包 DP 解决本题了。从左往右枚举每个贪心出来的 ps，记 dp_i 表示当前用掉了 i 个 o 时的最优答案。考虑枚举当前的 ps 用掉几个 o，它既要满足 $i+j$ 不超过当前 ps 左边 o 的数量，也要满足 $j \le$ (o 的总数/当前是从右往左第几个 ps) 即性质 2。

时间复杂度是 $O\left(n^2 \sum_{i=1}^{n} \frac{1}{i}\right) = O(n^2 \log n)$。

N Lament Rain

N.1 方法一

令 $N = 10^{12}$。考虑求 $p \le n, q \le m$ 内有多少 k-进制下的有限小数。

先将进制数分解为 $k = \prod_{i=1}^{\omega(k)} p_i^{\alpha_i}$，并记 k 的质因子集合为 $P = \{p_i\}$，在 $k \le N$ 时有 $|P| \le 11$。方便起见，不妨记 $P(n) = \gcd\left(n, \prod_{p_i \in P} p_i^{\infty}\right)$。容易知道，既约分数 p/q 在 k-进制下是有限小数当且仅当 $P(q) = q$，满足这个条件的 q 的数量至多为 N 以内 31-smooth 数的数量 $\Psi(N, 31) \approx 2.31 \times 10^6$。对于某个给定范围内的 k-进制下的有限小数 p'/q'，让其在 $q = P(q')$ 处作出贡献。记 q 处所求的贡献为 $f(q)$，容易写出：

$$f(q) = \sum_{\substack{c \le \lfloor m/q \rfloor \\ \gcd(c,k)=1}} \left\lfloor \frac{n}{c} \right\rfloor$$

考虑对 $\gcd(c, k) = 1$ 容斥，这相当于莫比乌斯反演。记

$$g(q, d) = \sum_{c \le \lfloor m/dq \rfloor} \left\lfloor \frac{n}{dc} \right\rfloor$$

就有：

$$f(q) = \sum_{d|k} \mu(d) g(q, d) = \sum_{d | \prod_{p \in P} p} (-1)^{\omega(d)} g(q, d)$$

对于某个特定的 d，预处理 $\lfloor n/dc \rfloor$ 在数论分块后每一段的取值与前缀和。结合 m/dq 关于 q 的单调性可以在均摊 $O(1)$ 的时间内求出每个 $g(q, d)$。数论分块的时间复杂度实际上不会超过：

$$\sum_{d | \prod_{p \in P} p} \sqrt{\frac{N}{d}} = \sqrt{N} \prod_{p \in P} \left(1 + \frac{1}{\sqrt{p}}\right) \le 24\sqrt{N}$$

对于某个特定的 d，$g(*,d)$ 只有 $\sqrt{N/d}$ 种取值。对于每种取值 $g(*,d)$，只需求有多少个 q 取到该值，相当于询问某个值域区间内有多少个满足条件的 q。提前预处理对 $\lfloor m/q \rfloor$ 数论分块时各段中有多少个 q，维护前缀和即可 $O(1)$ 回答询问。答案即为：

$$\sum_{\substack{q\le m\\ P(q)=q}} f(q)=\sum_{d\mid\prod_{p\in P}p}(-1)^{\omega(d)}\sum_{\substack{q\le m\\ P(q)=q}} g(q,d)$$

时间复杂度 $O(24\sqrt{N}+|P|\Psi(N,31))$。

Fun Fact 之一：听好了：本题原名 Pentiment，源出玻璃碴子大乱斗第四代，而如今玻璃碴子大乱斗已经更新到第六代了。

Fun Fact 之二：我们不幸发现给出的十二位数样例似乎并不够强，仍有选手在本题产生了多次 Wrong Answer 提交。

N.2 方法二

这里我们继续沿用方法一的记号。不妨考虑将分母 q 分解成 $P(q)$ 和 $q/P(q)$ 两部分，前者仅含有 k 中的质因子，而后者与 k 互质。p/q（不一定既约）是 k 进制下有限小数当且仅当 $\frac{q}{P(q)} \mid p$。

考虑先枚举所有的 $P(q)$，再枚举 $q/P(q)$，则有限小数的数量为

$$\sum_{j=1}^{m}[P(j)=j]\sum_{i=1}^{\lfloor m/j\rfloor}[\gcd(i,k)=1]\left\lfloor\frac{n}{i}\right\rfloor$$

此时我们考虑数论分块。我们需要对 $\lfloor n/i \rfloor$ 和 $\lfloor m/j \rfloor$ 分别分块，将分界点归并起来。在每一块里面，$\lfloor n/i \rfloor$ 和 $\lfloor m/j \rfloor$ 分别都是个常数。

我们不妨考虑将 $\lfloor m/j \rfloor$ 从小往大扫，很容易发现内层和式是 $[\gcd(i,k)=1]\lfloor n/i \rfloor$ 这个关于 i 的数列前缀和。考虑如何快速计算这个前缀和。由于数论分块时将 $\lfloor n/i \rfloor$ 看做常数，所以只需要快速统计一个区间内 $\gcd(i,k)=1$ 的个数即可。当这个区间比较大的时候，我们可以对 k 的质因子容斥，单次容斥的复杂度是 $2^{\omega(k)}$。而区间比较小的时候，我们可以直接用类似筛法预处理出来从 $1\sim T$（T 是一个可优化的分界线）中和 k 互质的情况。这部分的复杂度是 $T\log\log T$。查询时直接数该区间有多少个互质的数即可。

我们来分析复杂度。容斥需要做的次数是 n/T。进一步地，还很容易使用 `bitset` 来优化常数（这部分读者可自行思考实现）。我们要使得 $\frac{T\log\log T}{64}=\frac{N\cdot 2^{\omega(k)}}{T}$，将 $\log\log T$ 放大成 $\log\log N$，可以解得 $T=8\sqrt{N\cdot 2^{\omega(k)}/\log\log N}$，这部分最优的复杂度是 $O\left(\frac{\sqrt{N\log\log N\cdot 2^{\omega(k)}}}{8}\right)$。

当然我们也需要统计区间中 $P(j)=j$ 的数量。由于满足条件的 j 只有 $\Psi(N,31)\approx 2.31\times 10^6$，所以你可以随便用你开心的方法来实现。

最终的总复杂度是 $O\left(\Psi(N,31)+\frac{\sqrt{N\log\log N\cdot 2^{\omega(k)}}}{8}\right)$。

注：由于是校赛，出题人比较仁慈，所以不需要使用 `bitset` 也可以通过。STD 大约能在 0.1 倍时限内通过。我们也很高兴看到大家使用各种不同的做法通过了本题。

O Toxel 与 Toxtricity

根据微分算子的性质，我们有：

$$\sum_{i=0}^{t} c_i e^{i\mathrm{D}} f(x) = g(x)$$

令 $h(\mathrm{D}) = \sum_{i=1}^{t} c_i e^{i\mathrm{D}}$。那么我们有：

$$f(x) = h^{-1}(\mathrm{D}) g(x)$$

首先需要求出 $h(\mathrm{D})$：

$$\begin{aligned} h(\mathrm{D}) &= \sum_{i=0}^{t} c_i e^{i\mathrm{D}} \\ &= \sum_{i=0}^{t} c_i \sum_{j=0}^{+\infty} \frac{(i\mathrm{D})^j}{j!} \end{aligned}$$

我们不妨考虑 $\sum_{i=0}^{t} c_i \sum_{j=0}^{+\infty} (i\mathrm{D})^j$ 这个多项式，它和原来那个多项式每一项系数只差一个常数 $\frac{1}{j!}$，所以我们可以先计算出新的多项式。该多项式等于 $\sum_{i=0}^{t} \frac{c_i}{1-i\mathrm{D}}$。

很容易用分治 NTT 在 $O(t \log^2 t)$ 内计算出来。

第三步需要求出 $h^{-1}(\mathrm{D})g(x)$，其实也是一个 FFT。我们把 $g(x)$ 的式子展开：

$$\begin{aligned} & h^{-1}(\mathrm{D}) g(x) \\ =& \sum_{i=0}^{+\infty} [x^i] h^{-1} \cdot \mathrm{D}^i \sum_{j=0}^{+\infty} [x^j] g \cdot x^j \\ =& \sum_{j=0}^{+\infty} [x^j] g \sum_{i=0}^{j} [x^i] h^{-1} \cdot \mathrm{D}^i \cdot x^j \\ =& \sum_{j=0}^{+\infty} [x^j] g \sum_{i=0}^{j} [x^i] h^{-1} \frac{j!}{(j-i)!} x^{j-i} \\ =& \sum_{j=0}^{+\infty} [x^j] g \sum_{i=0}^{j} [x^{j-i}] h^{-1} \frac{j!}{i!} x^i \\ =& \sum_{i=0}^{n} \frac{1}{i!} x^i \sum_{j=i}^{+\infty} [x^{j-i}] h^{-1} \cdot [x^j] g \cdot j! \end{aligned}$$

到这里就能很明显地看出是一个卷积形式了。

时间复杂度 $O(t \log^2 t + n \log n)$。